Threat Modeling

Frank Swiderski

30 July 2003

Microsoft

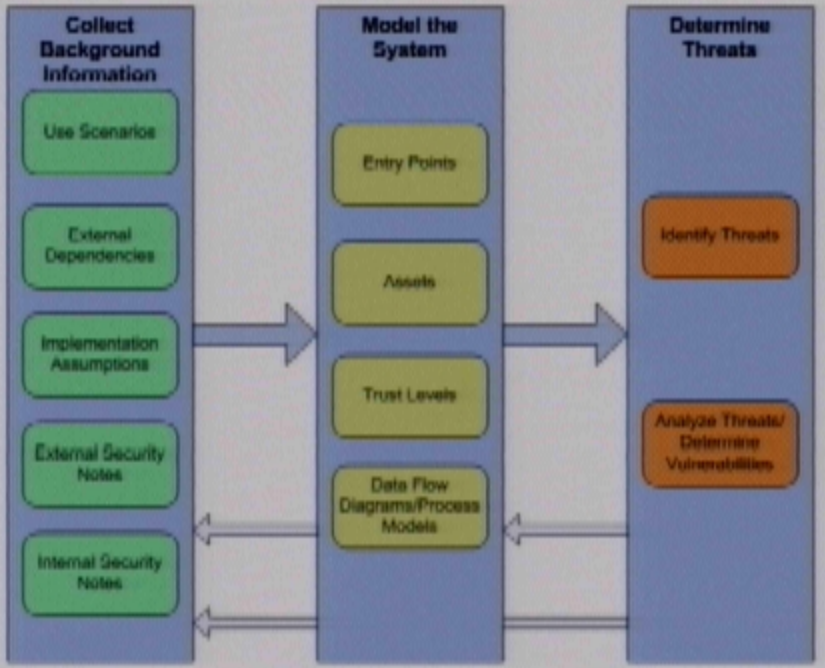
What is Threat Modeling?

- A process to understand and document security threats to a system that:
  - Is methodical and complete.
  - Will describe the system's threat profile.
  - Allows the security of the system to be characterized.
  - May find vulnerabilities.

Key Concepts

- A Threat Model describes a system's threat profile.
- A threat is not a vulnerability.
- The point of a threat model is more than just finding vulnerabilities.
- A system is anything that exposes functionality to an end user, and can describe anything from a single feature to a web application and its supporting infrastructure.

Threat Modeling Process



Collect Background Information

- Background information bounds the threat modeling discussion.
- It gathers information about dependencies that are security-critical.
- It provides necessary information for people to understand the threat model.

Identify Use Scenarios

- A use scenario explains how the system is intended or not intended to be used in deployment.
- Use scenarios help bound the threat modeling by describing the situations that were considered during the security design of the system.
- They can also explain situations where, if the system is deployed in an unsupported configuration, the security can be compromised.
- Use scenarios can be used as mitigation for threats to the system.

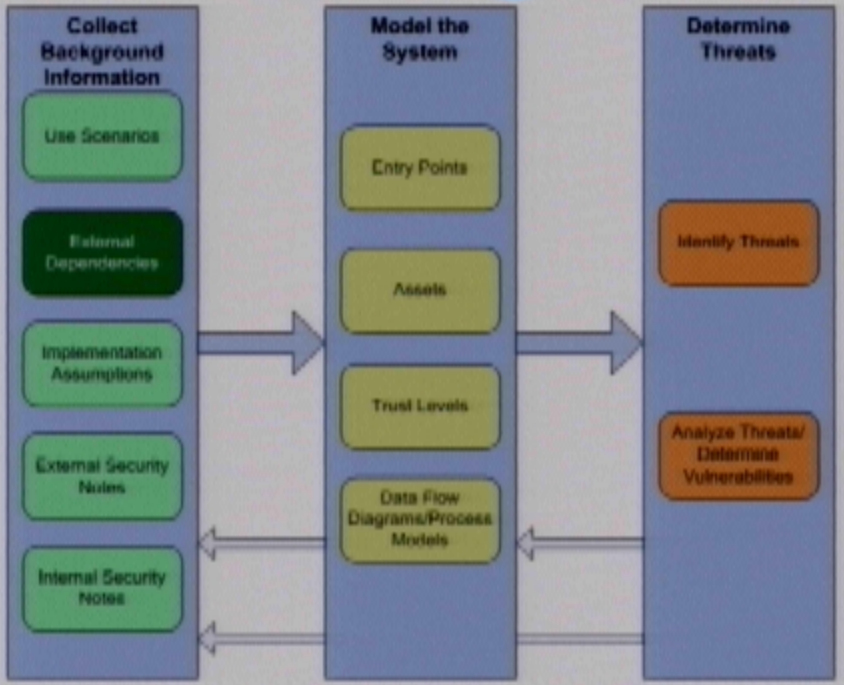
Tool: Use Scenarios Table

Use Scenarios	
ID	Description
1	The Fabrikam Phone 1.0 will be connected to the public switched telephone network. The security of this network is beyond the control of the Phone 1.0.
2	If the Phone 1.0 is installed in a location where untrusted users can access it, it should have local access control enabled.

Use Scenarios: Relevance

- **Who uses the information?** Use scenarios are used by the threat modeling team to limit the scope of the analysis. Managers and development leads must sign off on the individual use scenarios for the Threat Model to be valid. Later, the security test team can use these scenarios when conducting a penetration test, either to verify their validity or prove that they are not consistent with actual deployment.
- **How is the information collected?** The information is best provided by the designer of the system being modeled. If the system is a component that other teams are using, those teams may have input as to how the component is being used.
- **How is it used in the rest of the Threat Model?** Use scenarios can limit the discussion by describing scenarios that will not be considered (in other words, that are outside of the "safe" use of the system). During threat analysis, use scenarios can be used as the mitigation for conditions (for example, a condition may only be true if the system is used in an unsupported or "unsafe" scenario). Use scenarios may also help identify additional assets. For example, if a system is expected to run at a certain elevated privilege level, execution rights at that privilege level is an asset.

Threat Modeling Process



Identify External Dependencies

- External dependencies are requirements levied on systems outside of the system being modeled.
- They are dependencies on a certain behavior or specification compliance in an external system that, if broken, could cause threats in the system being modeled to manifest vulnerabilities.
- Often, these dependencies describe functions such as algorithm consistency across systems. For example, if two systems both normalize a string of text and take action based on the result, it is typically important that the normalized representation is the same across both systems.

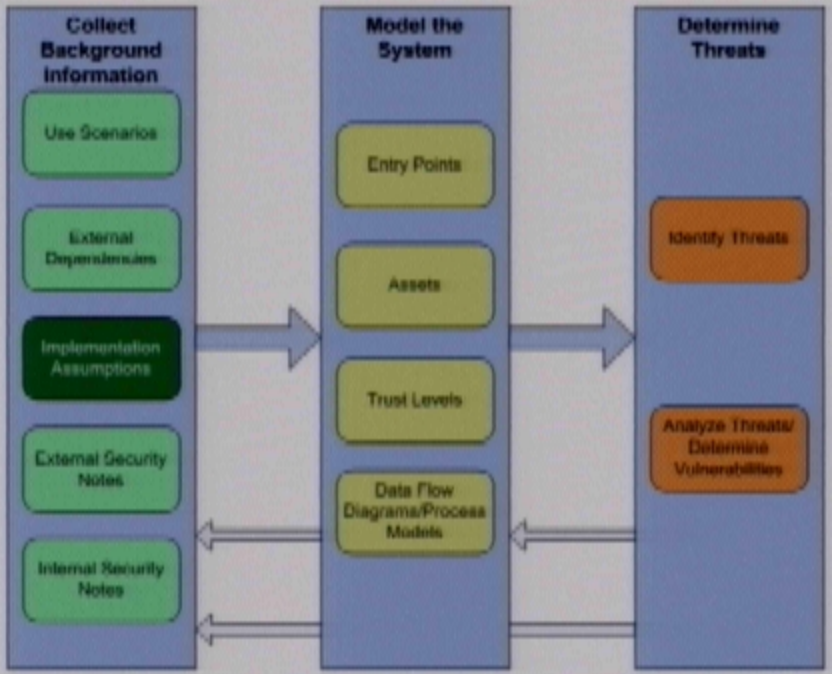
Tool: External Dependencies Table

External Dependencies	
ID	Description
1	The Fabrikam Phone 1.0 depends on the PSTN for providing power. There is a 2-day power cell in the Phone 1.0 that provides backup power should the power provided by the PSTN go down.

External Dependencies: Relevance

- **Who uses the information?** External dependencies are primarily used by the threat modeling team to validate assumptions between systems being modeled. The act of identifying, documenting, and investigating external dependencies can ensure that disparate systems and groups do not result in inconsistencies leading to vulnerabilities.
- **How is the information collected?** The designers and implementers of the system should be able to identify external systems that it depends on. Further, they can characterize the functionality of the external system that is used, thus providing a list of dependencies.
- **How is it used in the rest of the Threat Model?** External dependencies result in action items that must be resolved with the target system's team before a Threat Model can be considered valid. They validate cross-system assumptions that, if incorrect, might otherwise result in vulnerabilities.

Threat Modeling Process



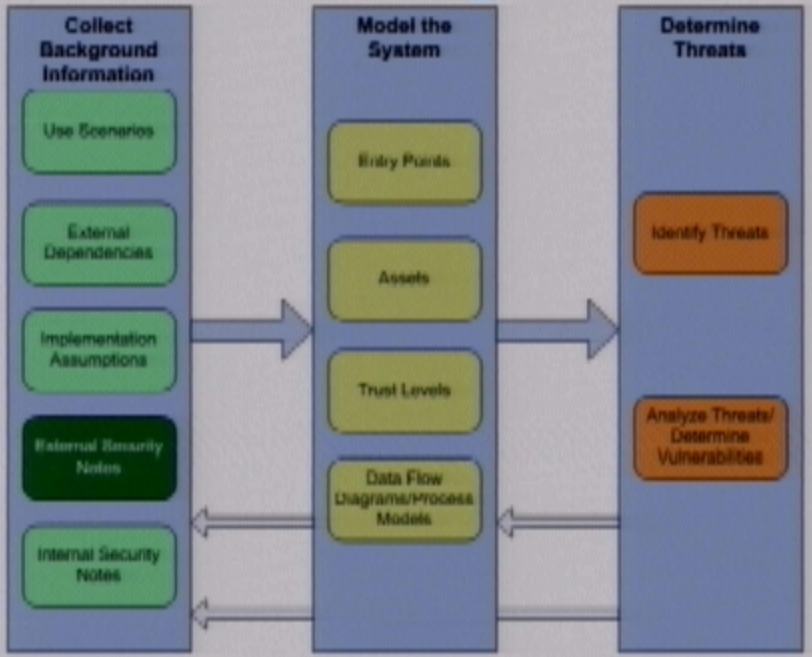
Identify Implementation Assumptions

- It is good to start the Threat Modeling process before a system is implemented.
- Implementation Assumptions are used when some or all of the system is in the design phase, and dictate specifics about how features must be implemented for the system to remain secure.
- Implementation Assumptions should be validated on completion of the implementation, in addition to revising the Threat Model as a whole to reflect the implementation.

Tool: Implementation Assumptions Table

Implementation Assumptions	
ID	Description
1	The voice-command dialing option has yet to be implemented. When this is added, it should not introduce a way to bypass current security features, such as long-distance call lockout.
2	If encrypted communication is added, key exchange should be done according to industry-accepted standards.

Threat Modeling Process



Identify External Security Notes

- External Security Notes are the counterpart to external dependencies.
- They provide security-relevant information to users that interface with the system being modeled.
- This information can be in the form of warnings against potential misuse that, while not constituting a vulnerability in the system being modeled, may surface a vulnerability in another system if it is not used correctly.
- Or, the information can be in the form of guarantees that the system makes for users.
- As an example, it may contain the specification for how filenames are normalized internal to the system.

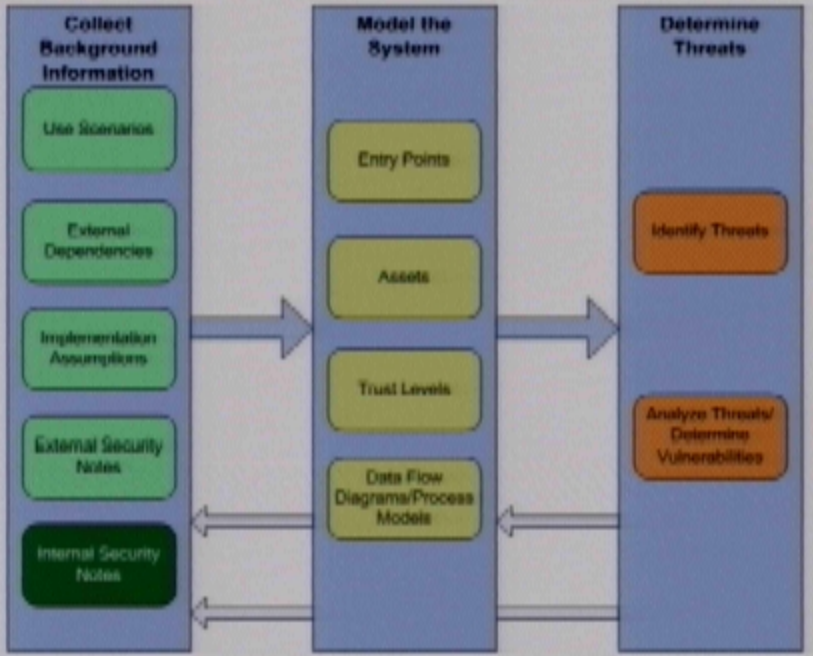
External Security Relevance

- **Who uses the informati**
  users whose systems in tur
  validate dependencies bas
  functionality to be added.
- **How is the informati**
  potential misuses o
  information should
  external depe
  information in
- **How i**
  are use
  s
  th

has a remote administration interface that has a default numeric
while the interface is disabled by default, the end user should ensure that
d is changed if it the feature is enabled.

end user wants to protect the speed dial list and whether the remote
nistration is enabled, he should enable local access control.

Threat Modeling Process



Identify Internal Security Notes

- Internal security notes are information that readers of the threat model should know to make the model more clear.
- They are often used to explain tradeoffs made in the design or implementation of the system that affect security.
- They should not be used as a replacement for threats and vulnerabilities.

Tool: Internal Security Notes Table

Internal Security Notes	
ID	Description
1	Speed dial information, messages, and the outgoing message are all stored in volatile RAM. The combination of volatile RAM and a battery backup for the Phone 1.0 is cheaper to manufacture than to use non-volatile RAM. This means, however, that power loss to the Phone 1.0 can cause loss of information if the battery backup is depleted.

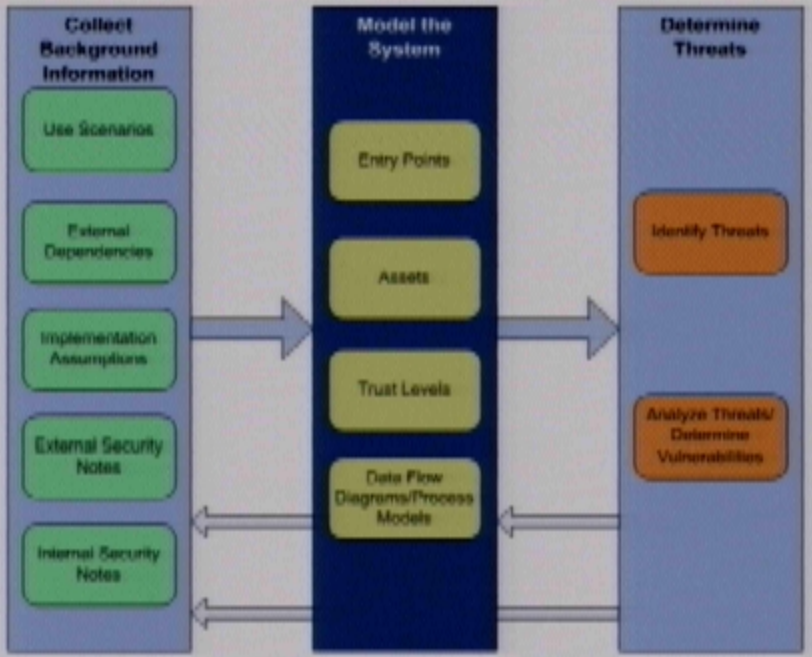
Internal Security Notes: Relevance

- **Who uses the information?** Internal security notes are used by reviewers of the threat model to understand security tradeoffs that were made in the design or implementation of the system.
- **How is the information collected?** Internal security notes are usually collected when the system is being modeled or when threats are being investigated. Often internal security notes come about when a threat exists because of a particular design decision, but that design decision was made to satisfy other, non-security requirements.
- **How is it used in the rest of the Threat Model?** Internal security notes are used when the threat model is reviewed for completeness. They are also used when mitigation for vulnerabilities is discussed.

Internal Security Notes: Relevance

- **Who uses the information?** Internal security notes are used by reviewers of the threat model to understand security tradeoffs that were made in the design or implementation of the system.
- **How is the information collected?** Internal security notes are usually collected when the system is being modeled or when threats are being investigated. Often internal security notes come about when a threat exists because of a particular design decision, but that design decision was made to satisfy other, non-security requirements.
- **How is it used in the rest of the Threat Model?** Internal security notes are used when the threat model is reviewed for completeness. They are also used when mitigation for vulnerabilities is discussed.

Threat Modeling Process



Model the System

- Modeling the system is critical to determining threats.
- It helps the threat modeling team understand the adversary's view of the system.
- It helps the team understand the internal workings of the system, allowing them to identify design- and implementation-specific threats.

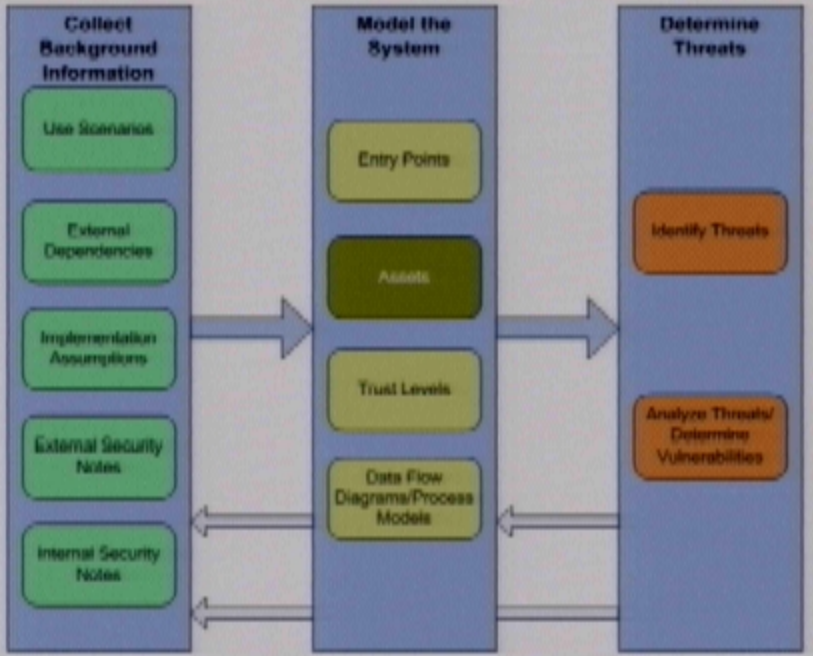
Identify Entry Points

- Entry Points define the boundary of the system being modeled.
- They list all places where the system consumes data from, provides data to, or performs actions on behalf of external entities.
- Entry points are often obvious: exposed APIs, listening sockets, RPC servers, Web Services, etc. Note however, that *any* place where the system interacts with external entities could be considered an entry point. For example, reading data from the file system is an entry point because the file system is likely not private to the system being modeled.

Identify Entry Points

- Entry Points define the boundary of the system being modeled.
- They list all places where the system consumes data from, provides data to, or performs actions on behalf of external entities.
- Entry points are often obvious: exposed APIs, listening sockets, RPC servers, Web Services, etc. Note however, that *any* place where the system interacts with external entities could be considered an entry point. For example, reading data from the file system is an entry point because the file system is likely not private to the system being modeled.

Threat Modeling Process



Identify Assets

- Assets are those things, both concrete and abstract, that could be targets of an attack by an adversary.
- Because of the widely varied functionality of systems, protected resources can also be widely varied. For example, a concrete example might be corporate data stored in a database. A more abstract example might be network coherency in a peer to peer application.
- Assets should be nouns.

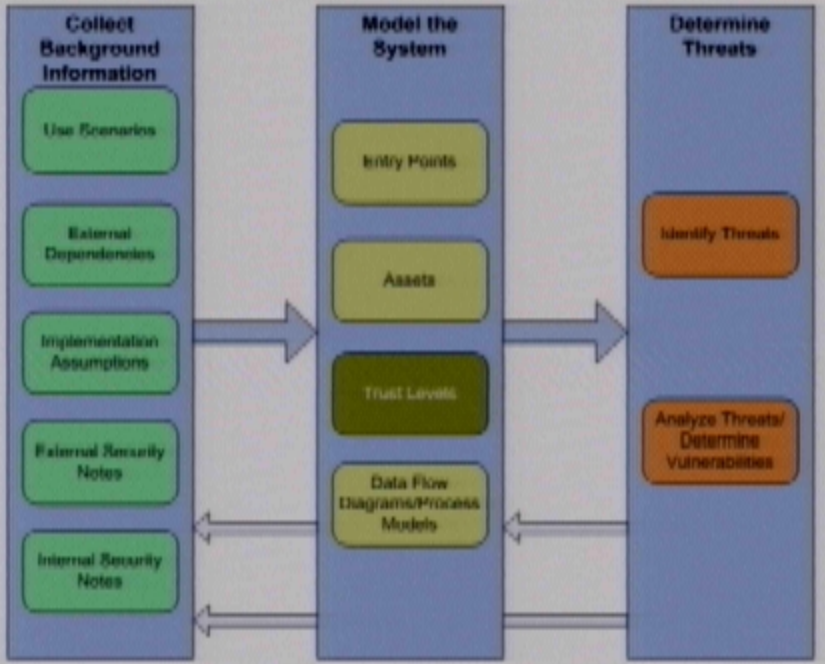
Tool: Assets Table

Assets			
ID	Name	Description	Trust Level
1	Speed-dial list	The speed dial list contains the names and numbers of often-used contacts.	(1) Administrator (2) Long-distance user (3) Local call user
2	Caller ID	Provides information about the incoming caller.	(1) Administrator (2) Long-distance user (3) Local call user
3	Access to the PSTN	The Phone 1.0 indirectly protects access to the PSTN.	(1) Administrator (2) Long-distance user (3) Local call user
4	Long-distance calling	The Phone 1.0 has optional lock-out for long distance calling so that only authorized users can make long distance calls.	(1) Administrator (2) Long-distance user
5	Phone configuration	The administrative configuration for the Phone 1.0	(1) Administrator
6	Messages	Messages left by callers when the Phone 1.0 has the answering machine feature enabled.	(1) Administrator (2) Long-distance user (3) Local call user
...			

Assets: Relevance

- **Who uses the information?** The Threat Modeling team uses the information when identifying threats. When analyzing security-critical processing, points where an asset is referenced should be scrutinized. The assets are, in essence, the targets of threats to the system. That is, a threat is what an attacker might try to do to or with an asset that would result in a violation of the systems expected security bounds.
- **How is the information collected?** Many assets are identified when discussing system functionality, use scenarios, and other background information. Questions to ask are: Does the system have access to any resources that an external entity would not normally have access to? What aspects of the system are critical to proper functionality?
- **How is it used in the rest of the Threat Model?** Assets are used during threat identification to identify an adversary's goals. Assuming that an adversary picks a protected resource as a target, what might he try to do to it?

Threat Modeling Process



Identify Trust Levels

- Trust levels characterize either entry points or assets.
  - In the case of entry points, they describe the external entity that can interface with the entry point.
  - For assets, they should indicate what privilege level would normally be able to access the resource.
- The type of trust level is specific to the entry point or protected resource. For example, some trust levels may correspond to NT groups. Other trust levels may simply describe what is known about the external entity (remote anonymous user, in the case of a public web server).

Identify Trust Levels

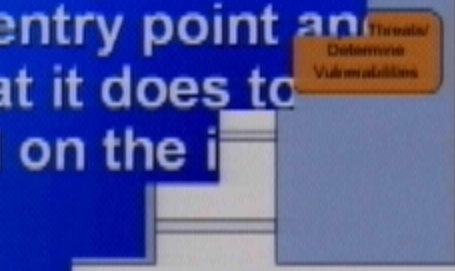
- Trust Levels that have more preconditions (such as requiring authentication) typically have a lower risk. The Trust Level table can be used to prioritize further discussion based on which categories pose the highest risk.

Tool: Trust Levels Table

Trust Levels		
ID	Name	Description
1	Administrator	The Phone 1.0 administrator has access to all features, and can bypass all security checks
2	Long-distance user	The Phone 1.0 can be configured to restrict long distance calling. The long-distance user is a phone user that is allowed to make long-distance calls.
3	Local call user	The local call user can only place outgoing local calls.
4	Denied user	The Phone 1.0 can be configured to not allow access to the phone without a password. The denied user is a user with no access.
5	Anonymous remote user	The anonymous remote user represents any data or incoming calls over the PSTN

Describe Processing on the Threat Path

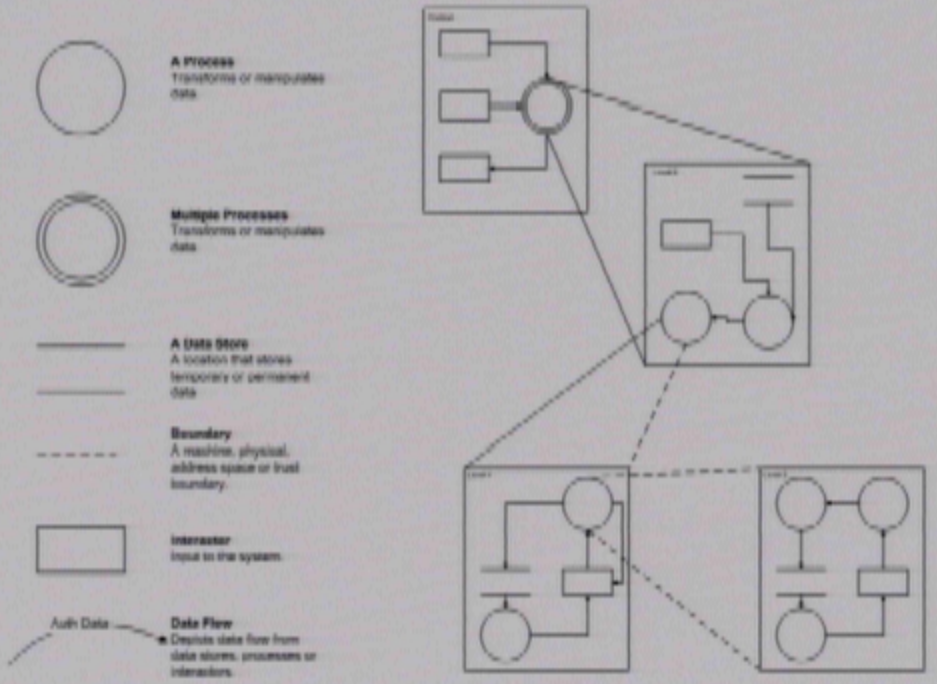
- The "Threat Path" is the processin
  that occurs based on the input t
  enumerated entry points.
- Follow processing from the
  component's entry point an
  determine what it does to
  data, or based on the i



Describe Processing on the Threat Path

- The "Threat Path" is the processing that occurs based on the input to the enumerated entry points.
- Follow processing from the component's entry point and determine what it does to the input data, or based on the input data.

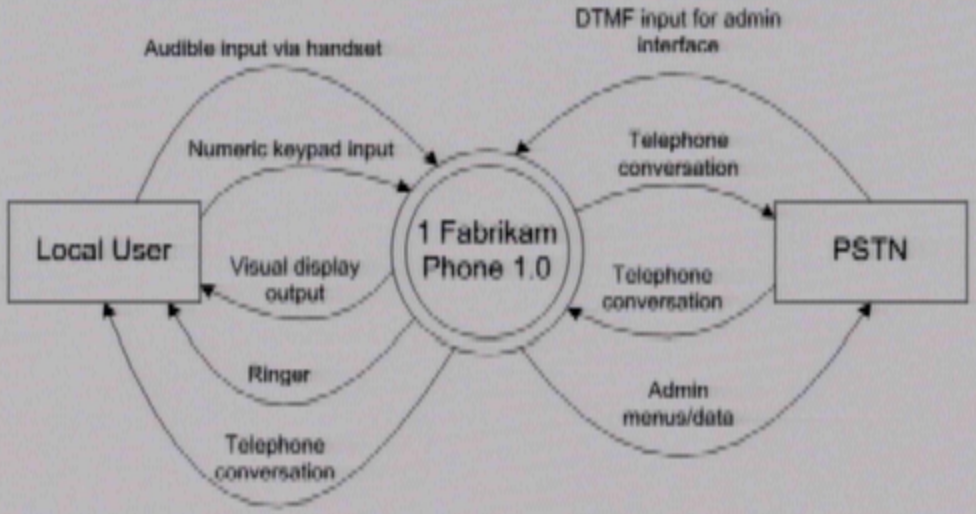
Tool: Data Flow Diagram



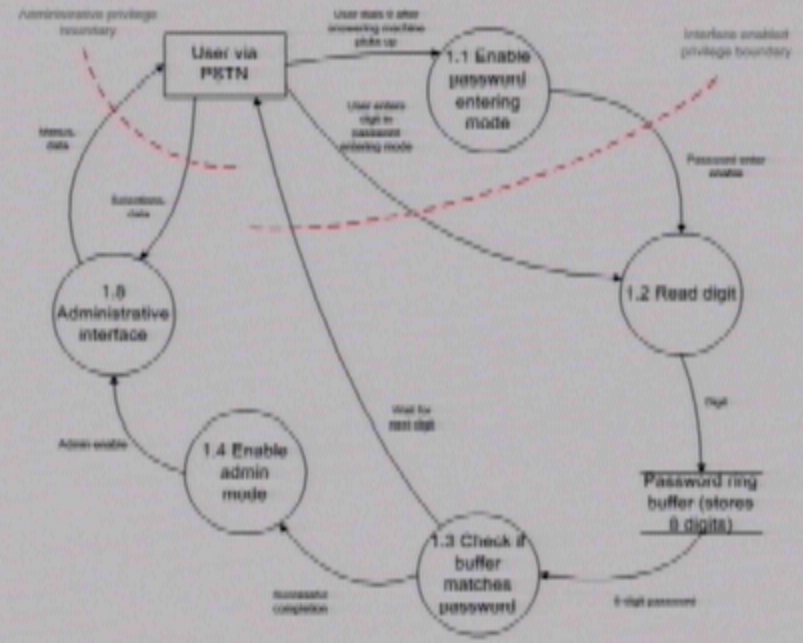
* Diagrams taken from Writing Secure Code, 2nd Edition

Microsoft

Tool: Data Flow Diagram



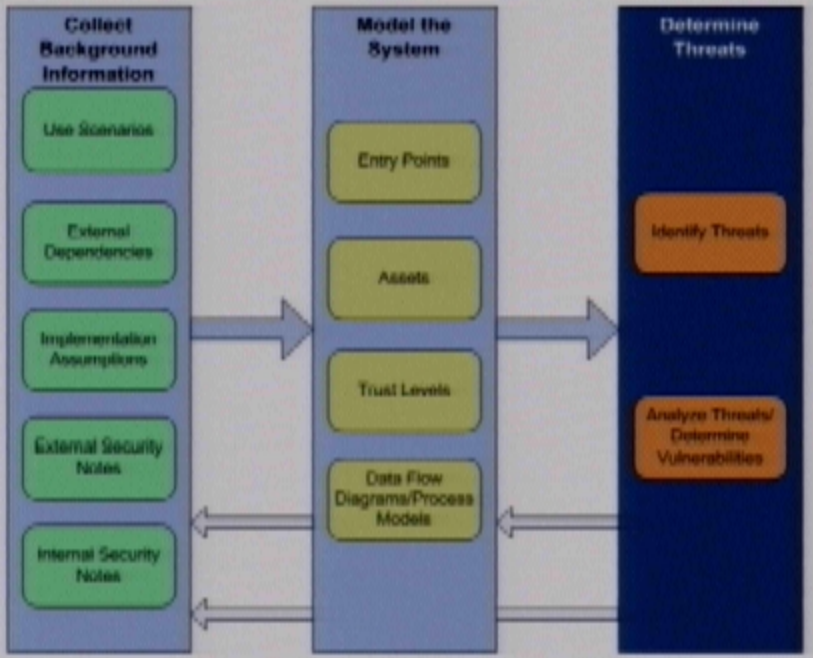
Tool: Data Flow Diagram



Data Flow Diagrams: Relevance

- **Who uses the information?** DFDs can be used by security testers to get a better understanding of the system's functionality and implementation. The visual representation of the data flows allows the tester to create attack hypotheses.
- **How is the information collected?** The designers and implementers of the system provide this information. It is often partially completed before any threat modeling meetings. During the meetings, however, they are usually expanded and more diagrams are created as the team analyzes the system.
- **How is it used in the rest of the Threat Model?** The DFDs are used during threat identification as a way to direct threat hypotheses. They allow the threat modeling team to better understand the functionality exposed by the system, and what an attacker's goals might be.

Threat Modeling Process



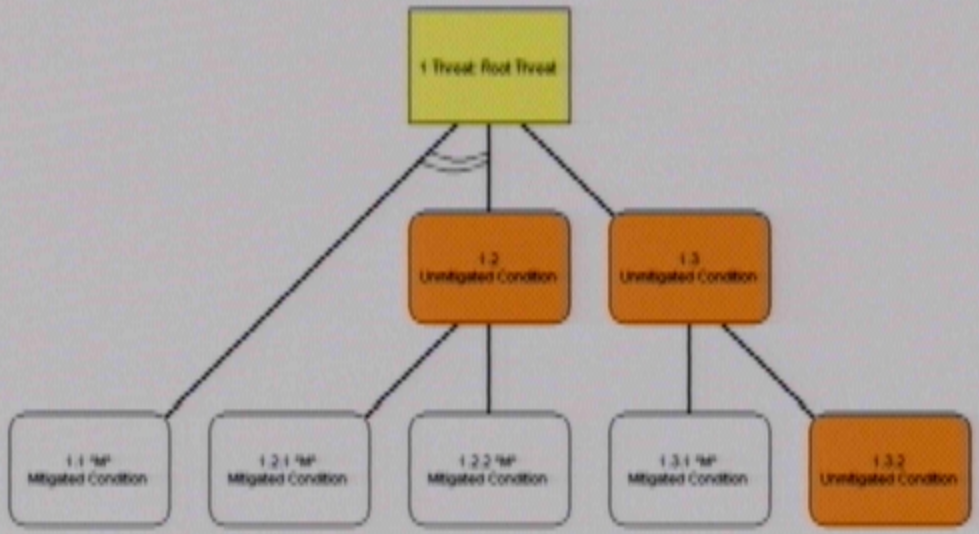
Determine Threats

- Enumerating threats creates a threat profile for a system, describing all of the potential attacks that should be mitigated against.
- Threats with valid attack paths are vulnerabilities.
- The security of a system can be expressed in terms of threats with appropriate mitigation vs. total threats, taking into account the severity of the threats with insufficient mitigation (vulnerabilities).

Enumerate Threats

- Identifying threats, as the critical point in creating a usable threat model, is appropriately the most difficult step in the process.
- The Threat Modeling team must take the information produced up to this point and create attack hypotheses.
- The team should not limit themselves to known vulnerabilities, rather, they should consider threats regardless of known mitigation.
- For a given entry point where a specific external entity interfaces with the system, what security-critical processing occurs, and what might a malicious external entity try to do to thwart that processing or otherwise use an asset outside of its expected use?

Tool: Threat Trees



Using STRIDE

- **STRIDE is used to classify the *effect* of threats.**
  - **Spoofing**. Spoofing allows an adversary to pose as another user, component, or other system that has an identity in the system being modeled.
  - **Tampering**. Tampering is the modification of data within the system to achieve a malicious goal.
  - **Repudiation**. Repudiation is the ability of an adversary to deny performing some malicious activity because the system does not have sufficient evidence to prove otherwise.
  - **Information Disclosure**. Information Disclosure is the exposure of protected data to a user that is not otherwise allowed access to that data.
  - **Denial of Service**. Denial of Service is when an adversary can prevent legitimate users from using the normal functionality of the system.
  - **Elevation of Privilege**. Elevation of Privilege is when and adversary assumes a Trust Level with different privileges than he currently has through illegitimate means.

* Some information taken from Writing *Secure Code*, 2nd Edition

Tool: Threats Table

Threats	
Threat	
ID	1
Name	Adversary gains access to the remote administration interface resulting in access to the phone configuration
Description	The Phone 1.0 has a remote administration interface that allows an authorized user to configure it via the PSTN. The interface is disabled by default, but can be enabled using the local keypad
STRIDE Classification	Tampering Information Disclosure Denial of Service Elevation of Privilege
Mitigated?	No
Known Mitigation	If the remote administration interface is enabled, the end user should change the default password
Investigation Notes	(none)
Entry Points	(6) Remote Administration (3) Telephone Line (2) Keypad
Assets	(5) Phone configuration

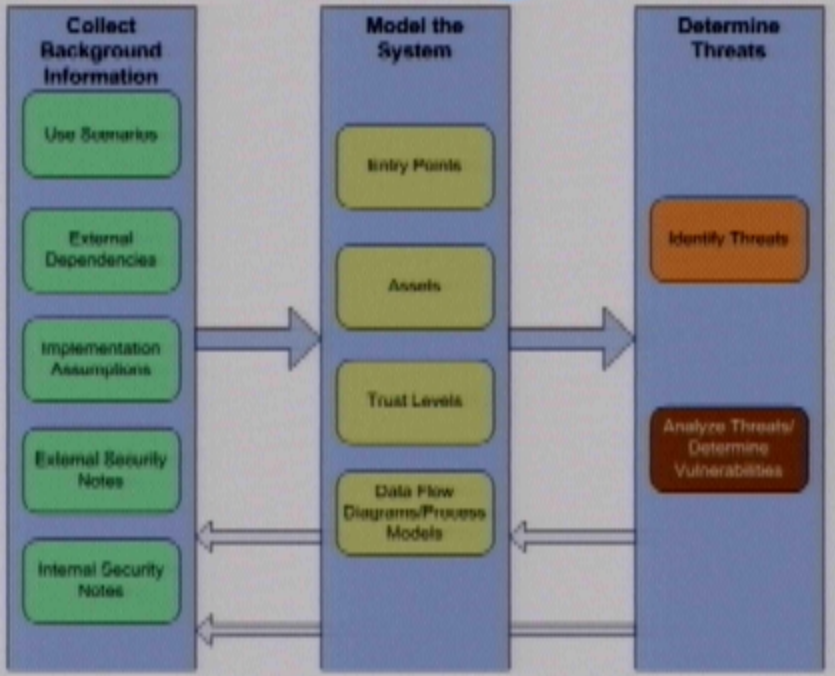
Tool: Threats Table

Threats	
Threat	
ID	2
Name	Adversary reads the speed dial list
Description	The speed dial list has sensitive information (names and telephone numbers).
STRIDE Classification	Information Disclosure
Mitigated?	No
Known Mitigation	Related Use Scenarios: (2) If the Phone 1.0 is installed in a location where...
Investigation Notes	(none)
Entry Points	(2) Keypad (4) Alphanumeric Display
Assets	(1) Speed-dial list

Threats: Relevance

- **Who uses the information?** Managers use threats to determine the security strengths of the system they are responsible for. Security testers use threats and their associated threat trees to test how well the system is resilient to attacks. Further, threats can be used as a plan of attack for a penetration test.
- **How is the information collected?** Persons responsible for the system's implementation and design are good sources. However, it is best to also include persons who did not work on the system during the threat modeling process. They are often able to think more critically about the system.
- **How is it used in the rest of the Threat Model?** Threats are later analyzed to determine if there are any vulnerabilities associated with them. They provide the basis for determining the strengths and weaknesses of the system.

Threat Modeling Process



Determine if Vulnerabilities Exist

- A Threat that has no (or insufficient) mitigating factors results in a Vulnerability—that is, something an attacker can exploit.
- For each Threat, determine if there are sufficient protections. Enumerate those that are Vulnerabilities.

Formulate Attacks

- Using threat trees
  - Threat trees start with what an attacker might try to do to or with a protected resource (threat), and create a tree of conditions that must be met in order obtain access to that protected resource.
  - Each condition can be translated to a test that can be performed programmatically or confirmed via code or design review. Conditions may or may not have mitigation, but do have DREAD ratings.
  - Access to one protected resource can facilitate access to others. Chaining threat trees can create more complex attacks ("attack chaining").